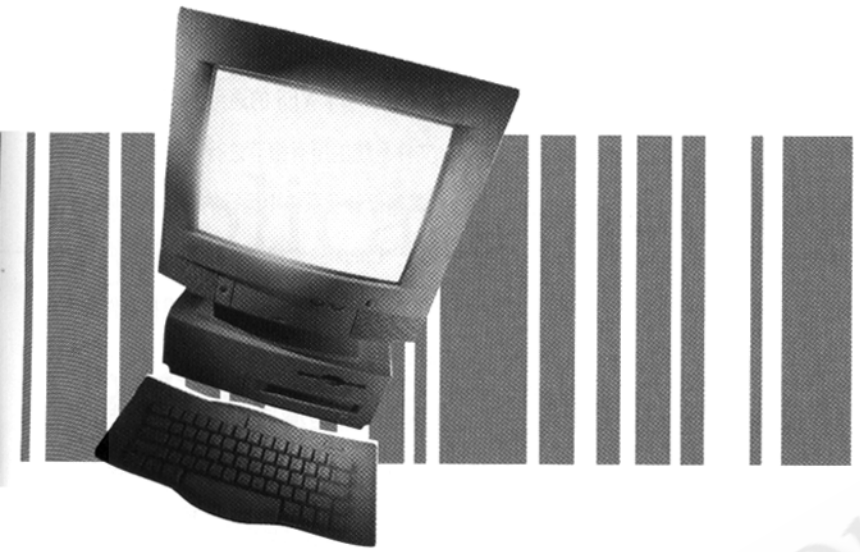

Unix/Linux 操作系统安全(三)

卿斯汉 (中科院信息安全技术工程研究中心 100080)

3 Unix/Linux系统内核的安全设计

UNIX核心的两个主要组成部分是文件子系统与进程子系统。文件子系统控制用户文件数据的存取与检索。在UNIX中，设备也是作为一种特殊的文件处理的。UNIX的文件子系统管理文件空间的分配，管理文件系统的空闲空间；进程控制子系统负责进程的同步、进程间通信、储存管理及进程调度。

3.1 文件子系统

3.1.1 文件系统概述

一个文件系统是由一系列块 (block) 构成，每个块的大小一般由生成这个文件系统时指定，或依赖于系统的实现。一个文件系统一旦生成，其中的块的大小是固定的。一个文件系统包含这样一些结构：引导块 (bootblock)、超级块 (super block)、索引节点表 (inode list)、数据块(data blocks)。引导块包含该文件系统的引导程序；超级块包含空闲索引节点表和空闲数据块表；索引节点用来存储文件相关信息及存储位置；数据块是磁盘上存放数据的磁盘块。

文件的逻辑结构和物理结构是十分不同的，逻辑结构是用户敲入cat命令后所看到的文件，用户可得到表示文件内容的字符流。物理结构是文件实际上如何存放在磁盘上的存储格式。用户认为自己的文件是连续的字符流，但实际上文件可能并不是以连续的方式存放在磁盘上的，长于一块的文件通常将分散地存放在盘上。然而当用户存取文件时，UNIX文件系统将以正确的顺序取各块,给用户提供文件的逻辑结构。

系统中的每一个文件都有一个与之相联系的索引节点，索引节点包括文件数据的磁盘地址明细表，它指出文件数据在磁盘上的存储位置，实现从物理结构到逻辑结构的转变。该表的结构由图4说明。

目录是一种文件，起到文件名到索引节点号之间转换的桥梁作用，也是使得文件系统成为树状结构的关键。目录文件的内容是由一系列的目录登记项构成，每项由该目录包含的一个文件名及该文件的索引节点号两部分组成。核心内部对文件是用索引节点来操作的，目录项便实现了这一从文件名到索引节点的转换。

3.1.2 设备文件

UNIX系统本系统上的各种设备之间的通信通过特别文件来实现，就程序而言，磁盘是文件，MODEM是文件，甚至内存也是文件。所有连接到

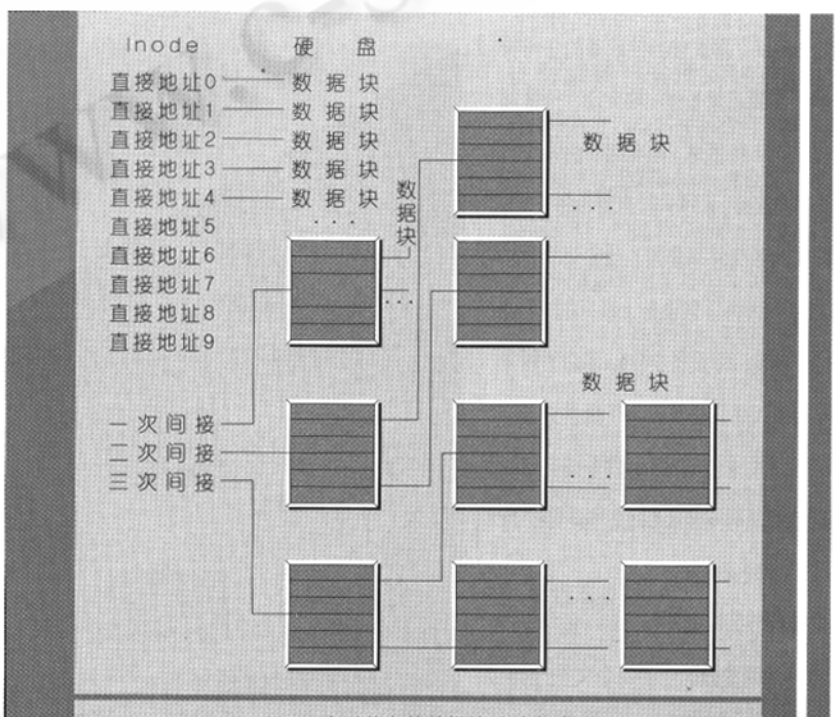

图4 索引节点的数据块地址存放

系统上的设备都在/dev目录中有一个文件与其对应，当在这些文件上执行I/O操作时，由UNIX系统将I/O操作转换成实际设备的动作。例如，文件/dev/mem是系统的内存。只有root能使用这个命令/etc/mknod建立设备文件，其参数是文件名、字母c或b分别代表字符特别文件或块特别文件、主设备号、次设备号。块特别文件是像磁带，磁盘这样一些以块为单位存取数据的设备，字符特别文件是如像终端、打印机、MODEM、或者其他任何与系统通信时，一次传输一个字符的设备，主设备号指定了系统子程序(设备驱动程序)，当在设备上执行I/O时，系统将调用这个驱动程序，调用设备驱动程序时，次设备号将传递给该驱动程序(次设备规定具体的磁盘驱动器,带驱动器,信号线编号,或磁盘分区)，每种类型的设备一般都有自己的设备驱动程序，文件系统将主设备号和次设备号存放在i节点中的磁盘地址表内，所以没有磁盘空间分配给设备文件(除i节点本身占用的磁盘区外)，当程序试图在设备文件上执行I/O操作时，系统识别出该文件是一个特别文件，并调用由主设备号指定的设备驱动程序，次设备号作为调用设备驱动程序的参数。

3.1.3 安全考虑

将设备处理成文件，使得UNIX程序独立于设备，即程序不必一定要了解正使用的设备的任何特性，存取设备也不需要记录长度、块大小、传输速度、网络协议等这样一些信息，所有细节由设备驱动程序考虑。要存取设备，程序只须打开设备文件，然后作为普通的UNIX文件来使用，从安全的观点来看这样处理很好，因为任何设备上进行的I/O操作只经过了少量的渠道(即设备文件)，用户不能直接地存取设备，所以如果正确地设置了磁盘分区的存取许可，用户就只能通过UNIX文件系统存取磁盘，而文件系统内部是有安全机制(文件许可)，所以，整个文件系统便是安全的。

但是，如果磁盘分区设备得不正确，任何用户都能够写一个程序读磁盘分区中的每个文件，对内存文件mem，kmem和对换文件swap也是如此，这些文件含有用户信息，一个"耐心"的程序可以将用户信息提取出来。要避免磁盘分区(以及其他设备)可读可写，应当在建立设备文件前先用umask命令设置文件建立屏蔽值。

3.1.4 虚拟文件系统(VFS)

现在Unix文件已经有了很大的变化，出现了更多有效的文件系统，为了支持各种文件系统，出现了VFS的概念，下面我们以Linux为例，Linux的最重要特征之一就是支持多种文件系统，这样它更加灵活并可以和许多其他种操作系统共存，Linux支持的文件系统有：ext, ext2, xia, minix, umsdos, msdos, vfat, proc, smb, ncp, iso9660, sysv,hpfs,affs以及ufs，毫无疑问,今后支持的文件系统类型还将增加。

Linux和Unix并不使用设备标志符(如设备号或驱动器名称)来访问独立文件系统，而是通过一个将整个文件系统表示成单一实体的层次树结构来访问它。Linux每安装(mount)一个文件系统时都会其加入到文件系统层次树中，不管是文件系统属于什么类型，都被连接到一个目录上且此文件系统上的文件将取代此目录中已存在的文件，这个目录被称为安装点或者安装目录。当卸载此文件系统时这个安装目录中原有的文件将再次出现。

VFS使得Linux可以支持多个不同的文件系统，每个表示一个VFS的通用接口，由于软件将Linux文件系统的所有细节进行了转换，所以Linux核心的其他部分及系统中运行的程序将看到统一的文件系统，Linux的虚拟文件系统允许用户同时能透明地安装许多不同的文件系统，虚拟文件系统的逻辑示意图如图5所示。

虚拟文件系统的设计目标是为Linux用户提供快速且高效的文件访问服务，同时它必须保证文件及其数据的正确性，这两个目标相互间可能存在冲突，当安装一个文件系统并使用时，Linux VFS为其缓存相关信息，此缓存中数据在创建、写入和删除文件与目录时如果被修改，则必须谨慎地更新文件系统中对应内容，如果能够在运行核心内看到文件系统的数据结构，那么就可以看到那些正被文件系统读写的数据块，描叙文件与目录的数据结构被不断的创建与删除而设备驱动将不停地读取与写入数据，这些缓存中最重要的是Buffer Cache，它被集成到独立文件系统访问底层块设备的例程中，当进行块存取时数据块首先将被放入Buffer Cache里并根据其状态保存在各个队列中，此Buffer Cache不仅缓存数据而且帮助管理块设备驱动中的异步接口。(未完待续)

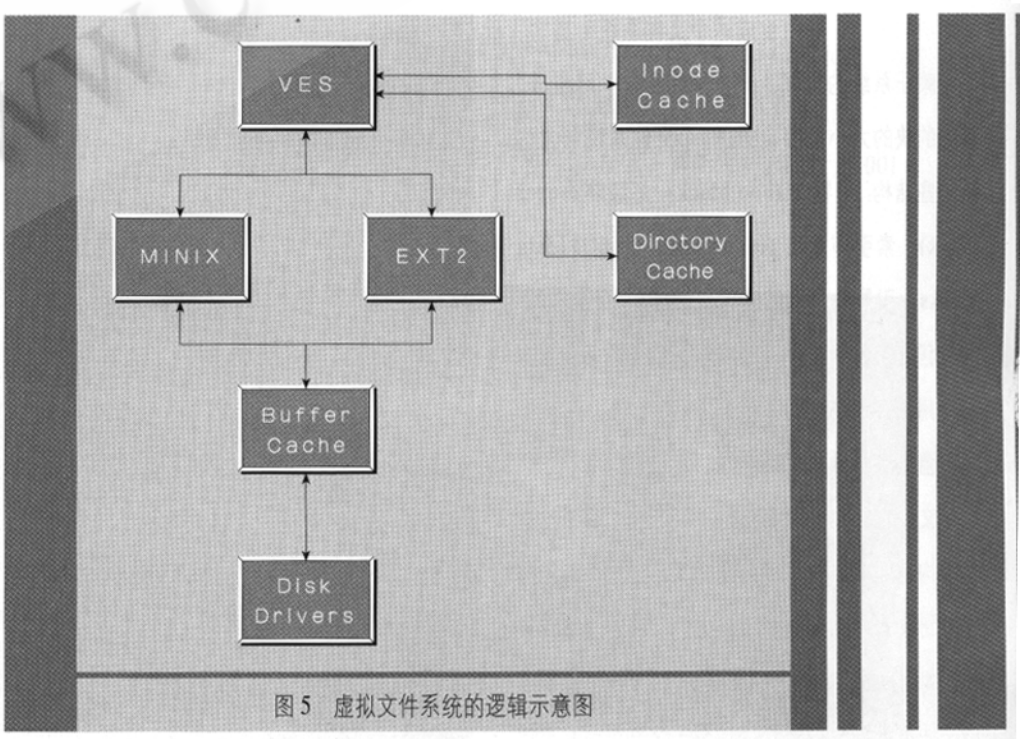

图5 虚拟文件系统的逻辑示意图